SIEMENS

PATENT
Attorney Docket No. 2002P18325WOUS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

| | | | | |
|---|---|---|---|---|
| Inventor: | M.-C. Fritsch et al. | ) | Group Art Unit: | 2611 |
| | | ) | | |
| Serial No.: | 10/538,152 | ) | Confirmation no: | 3257 |
| | | ) | | |
| Filed: | 06/08/2005 | ) | Examiner: | SINGH, Hirdepal |

Title: SYSTEM FOR THE GENERATION OF CODE

Mail Stop Appeal Brief - Patent
**COMMISSIONER FOR PATENTS**
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

## APPELLANT'S BRIEF UNDER 37 CFR 41.37

This brief is in furtherance of the Notice of Appeal filed in this application on 18 August 2009 in response to the final office action of 29 May 2009.

### 1. REAL PARTY IN INTEREST - 37 CFR 41.37(c)(1)(i)

The real party in interest in this Appeal is the assignee, Siemens Aktiengesellschaft.

### 2. RELATED APPEALS AND INTERFERENCES - 37 CFR 41.37(c)(1)(ii)

There is no other appeal, interference or judicial proceeding that is related to or that will directly affect, or that will be directly affected by, or that will have a bearing on the Board's decision in this Appeal.

## 3. STATUS OF CLAIMS - 37 CFR 41.37(c)(1)(iii)

| | |
|---|---|
| Claims pending: | 13, 17, 19, 23, 26, 31, 33-35 |
| Claims cancelled: | 1-12, 14-16, 18, 20-22, 24, 25, 27-30, 32 |
| Claims withdrawn but not cancelled: | None |
| Claims allowed: | None |
| Claims objected to: | 35 |
| Claims rejected: | 13, 17, 19, 23, 26, 31, 33-35 |

The claims on appeal are 13, 17, 19, 23, 26, 31, 33-35.

## 4. STATUS OF AMENDMENTS - 37 CFR 41.37(c)(1)(iv)

A. The last entered amendment was submitted 26 February 2009.

B. An amendment under 37 CFR 1.116 was submitted on 23 July 2009, but was not entered.

## 5. SUMMARY OF THE CLAIMED SUBJECT MATTER- 37 CFR 41.37(c)(1)(v)

Applicants' page, line, and paragraph numbers mentioned herein are relative to the substitute specification. Line counts do not include blank lines.

This invention relates generally to a system for generating industrial automation code for a manufacturing and/or processing plant from a description with control-relevant information. The description is described in a drawing based on a material flow in the plant, and includes a structure of the plant, know-how, and predecessor/successor relationships between components in the description representing elements of the plant. FIG 1 schematically illustrates a system according to aspects of the invention. FIG 2 illustrates a structure of controllers in an automation system. FIG 3 illustrates an example of port/information mapping and generation of automation code according to aspects of the invention. The invention is partially summarized in paragraph 10 (page 3, line 23 to page 4, line 19). The independent claims, 13, 26, and 33, are annotated individually below, relative to the specification and drawings.

**Independent claim 13** is directed to a system (FIG 1) for generating automation code for a manufacturing and/or processing plant (page 1, lines 15-16) from a description (FIG 1, element 1; and page 3, line 23 to page 4 line 19) enriched with control-relevant information (page 3, lines 23-25; and page 9, lines 15-19, "control-relevant information" is also called "automation-relevant information", as in page 8, lines 7-14), the system comprising:

a description comprising a drawing (page 5 lines 7-14) showing a layout of components (2) of the plant based on a material flow (page 5, line 26 to page 6, line 16) in the manufacturing and/or processing plant, wherein the drawing shows ports (FIG 1, element 6; page 4, lines 3-9) with control-relevant information (FIG 3; page 11, lines 17-19) for each component, and the drawing shows at least one functional module (3) for each component, wherein

input/output information is mapped to the ports, wherein the input/output information stems from directed relationships between the components (FIG 3 and page 11, lines 17-19), wherein the input/output information comprising predecessor/successor relationships among the components is included in the description (page 8 lines 19-24; and FIG 3; page 9 line 24 to page 10 line 12), wherein

signals (FIG 1, element 4; page 8 lines 24-29) provided for a transmission via the ports of the components are associated with each functional module and further comprising:

a first mechanism (FIG 1 element 5; and page 8, lines 3-4) for defining metainformation (page 4, lines 9-23) for the signals; and

a code generator (FIG 1 element 7; page 8, lines 5-7) for generating automation code by interconnecting the signals, wherein the automation code is generated on the basis of a structure of the plant (page 7, last two lines; page 10, lines 13-16) and know-how, including the predecessor/successor relationships, previously input into the description (page 4, lines 12-16; page 7, line 24 to page 8, line 6).

**Independent claim 26** is directed to a method for generating automation code for operating controllers (FIG 2; page 10 lines 13-16) in a manufacturing and/or processing plant (page 1, lines 15-16) from at least one description (FIG 1, element 1; page 3, line 23 to page 4 line 19) enriched with control-relevant information (page 3, lines 23-25; and page 9, lines 15-19; "control-relevant information" is also called "automation-relevant information", as in page 8, lines 7-14), the method comprising:

creating a description comprising a drawing (page 5 lines 7-14) of a layout of the plant, the layout representing components (2) of the plant by at least one respective functional block (3) or building block per component based on a material flow in the plant (page 5, line 26 to page 6, line 16), wherein the drawing comprises control-relevant information (FIG 3 and page 11, lines 17-19), and shows at least one port (FIG 1, element 6; and page 4, lines 3-9) for each component (2);

mapping input/output information regarding the ports between the components (FIG 3 and page 11, lines 17-19 ), wherein the input/output information stems from directed relationships including predecessor/successor relationships among the components contained in the descriptions (page 8 lines 19-24);

defining signals (FIG 1, element 4; and page 8 lines 24-29) associated with the functional blocks (3) or building blocks via the ports (6) of the components (2);

defining metainformation (page 4, lines 9-23) for the signals; and

generating automation code in a code generator (FIG 1 element 7; page 8, lines 5-7) for operating the controllers by interconnecting the signals (4), wherein the automation code is generated on the basis of a structure of the plant (page 7, last two lines; and page 10, lines 13-16) and know-how, including the predecessor/successor relationships, previously input into the description (page 4, lines 12-16; and page 7, line 24 to page 8, line 6).

**Independent claim 33** is directed to a system (FIG 1) for generating automation code for a manufacturing and/or processing plant (page 1, lines 15-16), the system comprising:

a plant description comprising a plurality of components (2), each component representing a given element of the plant (page 3, lines 25-27), each component comprising at least one function module (FIG 1, element 3) and at least one port (FIG 1, element 6; and page 4, lines 3-9), each port representing a connection point on the given element for data communication with another element of the plant, each function module being a reusable software object type that defines characteristics and functions of the given element (page 3, line 28 to page 4 line 3);

a communication network within the plant comprising a respective controller connected to each of the plant elements (FIG 2; page 10 lines 13-16);

the description comprising a drawing (page 5 lines 7-14) showing a layout of the components based on a flow of material in the plant (page 5, line 26 to page 6, line 16), the description further comprising control-relevant information (page 3, lines 23-25; and page 9, lines 15-19; "control-relevant information" is also called "automation-relevant information", as in page 8, lines 7-14) comprising rules that specify all allowable relationships including predecessor/successor relationships among the plant elements (page 8 lines 19-24; and FIG 3; page 9 line 24 to page 10 line 12), including allowable information content and flow directions among the ports (page 4 lines 17-19); and

a code generator (FIG 1, element 7; and page 8, lines 4-7) that automatically generates automation code for the plant that controls information flows among the controllers based on the drawing and the control-relevant information of the description, wherein the automation code is generated on the basis of a structure of the plant (page 7, last two lines; page 10, lines 13-16) and know-how, including the predecessor/successor relationships, previously input into the description (page 7, line 24 to page 8, line 6; page 9, lines 20-23; and page 10, lines 13-16).

## 6. GROUNDS OF REJECTION TO BE REVIEWED UPON APPEAL - 37 CFR 41.37(c)(1)(vi)

A. All claims 13, 17, 19, 23, 26, 31, and 33-35 are rejected based on rejection of the independent claims 13, 26, and 33 under 35 USC 112 second paragraph as omnibus claims.

B. All claims 13, 17, 19, 23, 26, 31, 33-35 are rejected based on rejection of the independent claims 13, 26, and 33 under 35 USC 112 second paragraph as being indefinite regarding what is included in the limitation "a drawing".

C. All independent claims 13, 26, and 33 are rejected under 35 USC 103(a) as being unpatentable over Burgess (US 5,805,896), in view of Sakurai et al. (US 6,334,076), Juras et al. (US 2002/0165744), and Elmqvist ("A Uniform Architecture for Distributed Automation", Advances in Instrumentation and Control, Instrument Society of America, Research Triangle Park, NC US, Vol. 46, Part 2, 1991, Pages 1599-1608).

D. Claims 13, 17, 19, 23, 26, 31, 33, and 34 are rejected under 35 USC 103(a) as being unpatentable over Burgess, in view of Sakurai, Elmqvist, and Leisten et al. (US 6,023,702).

## 7. ARGUMENT 37 CFR 41.37(c)(1)(vii)

Response to 35 USC 112 rejections of all claims 13, 17, 19, 23, 26, 31, 33-35 based on rejection of the independent claims 13, 26, and 33 under 35 USC 112 second paragraph as omnibus claims.

A. Claims 13, 26, and 33 cannot be considered omnibus claims. An omnibus claim reads as follows: "A device substantially as shown and described". This is the definition of an omnibus claim per MPEP2173.05(r), and is nothing like the subject claims. For example, claim 13 has a preamble and 5 clauses, and each clause recites one or more limitations. The claim does not incorporate a figure by reference.

A clause in each independent claim recites "a drawing" that is defined by some or all of the following exemplary limitations:

- shows a layout of components of the plant
- is based on a material flow in the plant
- shows ports with control-relevant information for each component
- shows a functional module for each component.

The application illustrates aspects of the claimed drawings as required for inventions capable of being illustrated. However, the claims do not recite "a drawing as shown", or "a drawing as shown in FIG 1". Instead, they recite defining limitations as listed above.

Response to rejections of all claims 13, 17, 19, 23, 26, 31, 33-35 based on rejection of the independent claims 13, 26, and 33 under 35 USC 112 second paragraph as being indefinite regarding what is included in the limitation "a drawing".

B. The limitation "a drawing" is defined within the claims as described above. The claim elements are referenced to the application specification and drawings in the section herein titled SUMMARY OF THE CLAIMED SUBJECT MATTER- 37 CFR 41.37(c)(1)(v).

The mere recitation of a "drawing" in a claim does not make the claim indefinite, nor has the Examiner cited any authority to support that position. Even if the content of the drawing were unspecified, the mere existence of a drawing is a specific limitation in and of itself. Beyond that, the content of the drawings of the present claims is defined with additional specificity within the claims themselves; e.g. claim 13 requires "a drawing showing a layout of components of the plant based on a material flow in the manufacturing and/or processing plant, wherein the drawing shows ports with control-relevant information for each component, and the drawing shows at least one functional module for each component".

Examiner interpreted claims 13, 26, and 33 "as the plant information related to layout is input into the system by a user" (Office Action 29 May 2009, top of page 4). Accordingly, in the non-entered response under 37 CFR 1.116 filed on 23 July 2009, Appellant attempted to amend the claims to explicitly recite that the plant layout is input into the system by a user, per Examiner's interpretation. Strangely, Examiner refused entry of this amendment on grounds that it changes the scope of the claims. However, Examiner has already examined and rejected the claims based on this interpretation. Should the Board overturn the pending prior art rejections, the Appellant would be agreeable to the entry of the previously submitted amendments under Rule 1.116 in order to satisfy the concerns of the Examiner.

Response to rejections of claims 13, 26, and 33 under 35 USC 103(a) as being unpatentable over Burgess, in view of Sakurai et al., Juras et al., and Elmqvist.

C. The arguments below apply to each of the independent claims 13, 26, and 33.

I. Predecessor/successor relationships: The independent claims 13, 26, and 33 each recite the generation of plant automation code by graphically mapping input/output information to ports on components based on predecessor/successor relationships contained in a description of the components. Burgess does not include predecessor/successor information in his component objects. Instead his directed relationships are defined in a graphical mapping stage by a programmer.

Examiner asserts on page 5 lines 3-5 of the office action of 05-29-2009 that Burgess in column 2, lines 23-30 discloses sending messages through ports. However, ports are not mentioned in these lines of Burgess, which only abstractly describe message information being passed between event objects.

Examiner asserts on page 5, line 15 that Burgess teaches directed connections of components producing program code, and the developer is shielded from the details thereof. However, this simply means that visual programming can be done, but that is not the issue. The developer still has to define directed relationships graphically, because they are not previously input into the description. Burgess col. 3, lines 29-34, lines 54-57 (FIG 4), and col. 4, lines 1-16 (FIG 5) describe visual programming as illustrated in FIG 4, in which a programmer graphically configures the directed relationships. Since predecessor/successor relationships are not contained in the descriptions of the objects 410, 420, 450, and 460, nothing prevents a reversal of the C-to-F and F-to-C calculator objects by the visual programmer, which would produce incorrect relationships.

> Burgess col. 3, lines 21-38: "FIG. 4 is a diagram illustrating visual programming of the present invention. To generate a visual program to implement a temperature

> converter, a programmer would position a Fahrenheit scroll bar 401, a Fahrenheit display 430, a Centigrade scroll bar 420, and a Centigrade display 440. The programmer would also position an FtoC calculator 460, which converts a Fahrenheit value to a Centigrade value, and a CtoF calculator 450, which converts a Centigrade value to a Fahrenheit value. In one embodiment, the components are selected from an extendible list of available components. The programmer then connects the components through their ports. The connections 412->461 and 412->431 indicate that when the Fahrenheit scroll bar is changed (e.g., slider moved), the new value is sent to the FtoC calculator and the Fahrenheit display. The connection 462->421 indicates that when the FtoC calculator calculates a new Centigrade value, the new value is sent to the Centigrade scroll bar."

In contrast, Applicants' predecessor/successor relationships are already contained in a description of each component, thus, they are predetermined, constraining the connections to a proper order by allowing fewer degrees of freedom in order to reduce the possibility of error.

> Applicants' paragraph 10, lines 1-3: "In the system according to the invention data continuity is achieved in that control-relevant information is already contained in a description."

> Applicants' paragraph 20, lines 13-18: "The information already contained in the description 1 is used to allocate input and output information to the ports. The predecessor-successor relationships between the components are governed by this information, i.e. who sends data to whom via which data input is defined thereby."

This prevents incorrect relationships that are possible in Burgess, because the predecessor-successor relationships are defined prior to graphical mapping stage for local plant code generation.

> Applicants' paragraph 21: "On the basis of the metainformation, the components 2 are connected to one another by automated means. Particular connections between the components 2 can only be implemented if this is permitted by the constraints described in the metainformation. Automated "wiring" of the components 2, and therefore automatic generation of automation code, are therefore effected."

Applicants' predetermined predecessor/successor relationships restrict freedom in order to reduce complexity in automation programming, because incorrect options are eliminated from consideration. Continuity of expert information and earlier know-how guides and limits the plant automation code developer.

> Applicants' paragraph 22, all lines: "The work of the development engineer is greatly facilitated thereby, since fewer degrees of freedom exist as a result of the definition of the metainformation, reducing the possibilities of error. In addition, a continuous information flow is ensured, reducing the loss of already established know-how during the development of the automation system."

> Applicants' paragraph 34, lines 2-3: automation code is generated on the basis of existing descriptions 1 of a plant structure.

II. Previously input plant structure and know-how: On page 6, lines 4-7 of the office action of 05-29-2009, Examiner concedes that Burgess does not teach code generated on the basis of a structure of a manufacturing or processing plant and know-how including predecessor/successor relationships previously input into the description. These elements are recited in all of the independent claims of this application, including predecessor/successor relationships, as taught in paragraphs 22-25 of the specification. This is not simply an intended use, because it prevents sequence errors.

Sakurai provides standard program modules previously coded by program engineers. These standard modules are independent of plant type (col. 4, lines 3-6). A plant operator inputs a plant operation procedure by keyboard, which includes a time sequential control flow (col. 3, lines 61-63). This results in a selection of standard program source code modules for assembly into a load module for execution in a programmable logic controller (PLC). There is no teaching that the standard program modules include a previously entered description of predecessor/successor relationships for plant components.

Examiner cites Sakurai's sequential flow chart (SFC) and block diagram as evidence of automatic code generation using a drawing of a plant layout with automatic predecessor/successor control as claimed by Applicants. However, the operator of Sakurai must select appropriate program modules using a module identification code, must then input specifications of a plant operation procedure to modify each program module, and must specify their execution order and interconnections. This is extremely more complex than Applicants' code generation as claimed. Sakurai requires an operator to have a systems level understanding of program modules, load modules, symbolic address resolution, linking, and loading.

> Sakurai col. 5, lines 30-41: "The plant operation procedure is inputted to the CRT controller 20 preferably in the form of the sequential flow chart (hereinafter called SFC) 52 and arithmetic block diagram 50, by using the keyboard 5. In the CRT

> controller 20, while observing CRT 21, an operator inputs an identification code of a module through the keyboard 5 to thereby read the module from the stacker 156 via the system 10C. While observing the read-out plant operation procedure information, the operator enters from the keyboard the program generating information necessary for the generation of a program, i.e., program module execution level and order, signal interconnection or the like."
>
> Sakurai col. 6, lines 49-54: "(3) The interconnection relationship between customized modules are represented not by absolute addresses of a memory storing the modules but by the device numbers defined by various list information. After customized modules are combined and edited as a load module, address translation is carried out."

III. Description in drawing: Examiner concedes on page 6, lines 7-8 that Burgess does not teach components described in a drawing comprising control-relevant information in a manufacturing and/or processing plant. On page 7 of the office action Examiner cites Sakurai col. 2, lines 20-51. However, these lines do not mention a drawing or graphics at all. Examiner also cites Sakurai col. 4, lines 8-22. However, these lines describe visually monitoring of a PLC program while it is executing, in order to verify proper program operation -- not graphically configuring a specification for automatic code generation.

IV. Description based on material flow: Examiner concedes on page 6, line 8-10 that Burgess does not teach control information described in a drawing based on a material flow in a manufacturing and/or processing plant. Examiner then asserts on page 8, lines 1-5 that Elmqvist supplies the claimed feature of drawings with control-relevant information based on material flow in a processing plant, because it is inherent that the physical objects of the plant form the path for the material or fluid flow as shown in the example of the tank system (figures 1-5). However, as in Burgess, this tank system layout only exists after the visual designer has selected the graphic components and placed them in this order. These graphical components do not have predecessor/successor descriptions stored in them to require an order based on a material flow. Instead, the design module definitions of Elmqvist are purely hierarchical (FIG 2). The first line under VISUALIZATION on page 1605 states: "The complete picture, as seen in a window, is a hierarchical picture according to the module hierarchy." The two tanks of FIG 1 are just instantiations of the same "tank" object. Thus their order in FIG 1 could be reversed -- the same kind of mistake as discussed for Burger above. It so happens that this reversal would not make a difference in FIG 1 of Elmqvist, but this is only because FIG 1 is too simple an example to illustrate an ordered relationship. The examples of Elmqvist and Burgess are both highly simplified, but at least the example of Burgess can be used to show how order is important, which is the case in a manufacturing or processing plant.

V. Predecessor/successor relationships in view of Juras: Examiner concedes on page 6, lines 4-7 that Burgess does not teach system components defined to have predecessor/successor relationships. Examiner then cites Juras, who teaches a process for strategic business planning, including developing quotes, creating business work plans, managing operations; and estimating a total cost of product development. Output of this process is a list of things to do (par. 51, FIG 14). This is a business planning tool, not a code generator for a manufacturing plant. It is unrelated to the present invention, which is not a business plan, but an automation code generator for a manufacturing plant.

> Juras [0035]: As shown in FIG. 1, the present invention is a product development process which takes development of a product from an initial Request For Estimate (RFE), through the creation of a product development or business plan to an implementation of the plan for a product launch. The product development process sets up a time schedule for delivering the contracted deliverables to the customer.
>
> Juras [0036]: The product development process uses tools embodied in software which provide a structured process for developing quotes, creating business and functional work plans, and managing operations to design and build manufacturing systems to manufacture and deliver the contracted product to the customer. The tools include a menu structure, plant layout, product development and process flow chart.

This process does not generate software, but generates requests for quotes, bills of material, cost estimates, project planning documents, manpower requirements, lists of deliverable work products, approval steps, and the like. Juras mentions plant layout only once, in paragraph 36 above, in relation to project planning for plant design. He says nothing about predefined predecessor/successor relationships among elements of a plant or generating software to operate a plant. Juras simply coordinates tasks done by humans, which does not apply to the present invention. Nowhere does he show a plant layout, either by a drawing or other means.

Response to rejections of claims 13, 17, 19, 23, 26, 31, 33, and 34 under 35 USC 103(a) as being unpatentable over Burgess, in view of Sakurai, Elmqvist, and Leisten et al.

D. Examiner cites Leisten as teaching predecessor and successor activities in a plant. However, Leisten teaches a method of project management of workgroups. His method has nothing to do with generating automation code for a manufacturing and/or processing plant, automatically or otherwise. His method has nothing to do with defining input/output signal connections between components for generation of automation code. He uses house building as an example of a managed project. His predecessor and successor activities simply refer to a calendar schedule of a human work project. Leisten simply coordinates tasks done by humans, which does not apply to the present invention.

> Leisten col. 15, lines 34-49: "The term "Work Process" is used in the following for the full integration of process and project management under the inventive concept. The example is taken from the context of a building enterprise in the building industry, where the building general contractor erects houses according to the requests of an owner."
>
> Leisten col. 15, lines 50-53: "For the process of building the standard house, a project schema 102 is defined which manages the application of resources, as building teams and building equipment within well defined schedule and calendar constraints."

This does not fill the deficiency of Burgess, noted by Examiner on page 17 lines 15-19, that Burgess doesn't specifically teach code generation for a manufacturing and/or processing plant, and that automation code is generated on the bases of a structure of the plant and know how, including predecessor/successor relationship previously input into the description.

8. CLAIMS APPENDIX - 37 CFR 41.37(c) (1) (viii).

A copy of the claims involved in this appeal is attached as a claims appendix under 37 CFR 41.37(c) (1) (viii).

9. EVIDENCE APPENDIX - 37 CFR 41.37(c) (1) (ix)

None is required under 37 CFR 41.37(c) (1) (ix).

## 10. RELATED PROCEEDINGS APPENDIX - 37 CFR 41.37(c) (1) (x)

None is required under 37 CFR 41.37(c) (1) (x).

Respectfully submitted,

Dated: Sept. 24, 2009

By: Janet D. Hood

Janet D. Hood
Registration No. 61,142
(407) 736-4234

Siemens Corporation
Intellectual Property Department
170 Wood Avenue South
Iselin, New Jersey 08830

# APPENDIX OF CLAIMS ON APPEAL

13. A system for generating automation code for a manufacturing and/or processing plant from a description enriched with control-relevant information, the system comprising:

a description comprising a drawing showing a layout of components of the plant based on a material flow in the manufacturing and/or processing plant, wherein the drawing shows ports with control-relevant information for each component, and the drawing shows at least one functional module for each component, wherein

input/output information is mapped to the ports, wherein the input/output information stems from directed relationships between the components, wherein the input/output information comprising predecessor/successor relationships among the components is included in the description, wherein

signals provided for a transmission via the ports of the components are associated with each functional module and further comprising:

a first mechanism for defining metainformation for the signals; and

a code generator for generating automation code by interconnecting the signals, wherein the automation code is generated on the basis of a structure of the plant and know-how, including the predecessor/successor relationships, previously input into the description.

17. The system according to claim 13, further comprising a mechanism for inputting control-relevant information for use in the description.

19. The system according to claim 13, wherein the material flow, and/or an energy flow, and/or an information flow in the plant is provided as a basis for mapping the predecessor/successor relationships between the components.

23. The system according to claim 13, wherein the generation of automation code is provided for central and/or distributed automation solutions.

26. A method for generating automation code for operating controllers in a manufacturing and/or processing plant from at least one description enriched with control-relevant information, the method comprising:

creating a description comprising a drawing of a layout of the plant, the layout representing components of the plant by at least one respective functional block or building block per component based on a material flow in the plant, wherein the drawing comprises control-relevant information, and shows at least one port for each component;

mapping input/output information regarding the ports between the components, wherein the input/output information stems from directed relationships including predecessor/successor relationships among the components contained in the descriptions;

defining signals associated with the functional blocks or building blocks via the ports of the components;

defining metainformation for the signals; and

generating automation code in a code generator for operating the controllers by interconnecting the signals, wherein the automation code is generated on the basis of a structure of the plant and know-how, including the predecessor/successor relationships, previously input into the description.

31. The method according to claim 26, wherein automation code is generated for central and/or distributed automation systems.

33. A system for generating automation code for a manufacturing and/or processing plant, the system comprising:

a plant description comprising a plurality of components, each component representing a given element of the plant, each component comprising at least one function module and at least one port, each port representing a connection point on the given element for data communication with another element of the plant, each function module being a reusable software object type that defines characteristics and functions of the given element;

a communication network within the plant comprising a respective controller connected to each of the plant elements;

the description comprising a drawing showing a layout of the components based on a flow of material in the plant, the description further comprising control-relevant information comprising rules that specify all allowable relationships including predecessor/successor relationships among the plant elements, including allowable information content and flow directions among the ports; and

a code generator that automatically generates automation code for the plant that controls information flows among the controllers based on the drawing and the control-relevant information of the description, wherein the automation code is generated on the basis of a structure of the plant and know-how, including the predecessor/successor relationships, previously input into the description.

34. The system of claim 33, wherein the network comprises at least two control zones, each control zone comprising a subset of the plant elements controlled by a respective subset of the controllers, and the network further comprises a coordinating controller for each control zone, and wherein the description describes a topology of the network for the automatic code generation.

35. The method according to claim 26, wherein the metainformation comprises one or more input/output parameters with a value "S" or "P" for each component, and wherein an algorithm operates the code generator to automatically generate code for connecting the components as follows:

```
for all components
    for all inputs of the respective functional module
        for all predecessors of the component
            a) search for a predecessor functional module that has an output parameter with a value "S";
            b) search for an input of the respective functional module that has a parameter with a value "P"; and
            c) connect the output of the predecessor functional module that has an output parameter with a value "S" to the input of the respective functional module that has the parameter with a value "P".
```

## EVIDENCE APPENDIX

None.

# RELATED PROCEEDINGS APPENDIX

None.